MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

G-

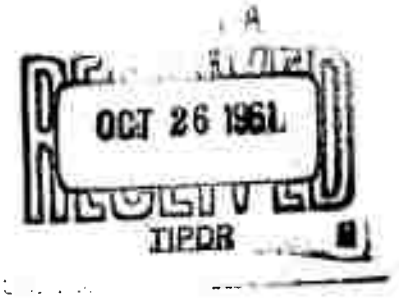


LEXINGTON MASSACHUSETTS

# A WEIGHT FORMULA FOR GROUP CODES

by

Gustave Solomon




Introduction:-

For n odd, let A be any (k, n) group code, i.e. a k-dimensional subset of $V_n(F)$, the vector space of dimension n over the field F of two elements. The weight of a vector $a = (a_i) \in A$, w(a), is defined to be the number of $a_i = 1$. A classical coding theory problem is to determine the minimum weight of all non-zero vectors in A. We are also interested in the values of all weights occuring in the code as well as the number of vectors belonging to a certain weight. This counting problem -- this looked-for formula or analytic expression for the weight of a vector in terms of the parameters which characterize the code is complicated by the fact that our code operations are all modulo 2 and the weight function is integer valued. We introduce in this paper the exponential weight of a vector $\beta^{w(a)}$ which is an element of a finite field over F. Using the general representation of group codes in Solomon [3], we obtain a formula for the exponential weight of a vector as a function of several independent variables,

the parameters of the general representation. The coefficients of the formula are obtained using elementary multiplication on the cyclic group of the $n^{th}$ roots of unity (this usually requires a computer for large n). With this formula we hope to make a fresh start on answering some of the long-standing problems of coding theory.

For L. B.

## 1. General Representation of Group Codes

We need the general representation of group codes via polynomials found in Solomon [3]. We reproduce the results here since they are essential to our work.

Let $a = (a_0, a_1, \ldots, a_{n-1})$ be a vector in $V_n(F)$, the vector space of odd dimension n over the field F of two elements $\{0, 1\}$. The components $a_i$ of the vector a can be considered as the values of a polynomial $g_a(x)$ for $x = \beta^i$, i.e., $a_i = g_a(x = \beta^i)$, where $\beta$ is a fixed primitive $n^{th}$ root of unity. This polynomial $g_a(x) = \sum_{i=0}^{n-1} c_i x^i$ † has the property that $g_a(x)^2 = g_a(x)$ for $x = \beta^0, \beta^1, \ldots, \beta^{n-1}$. Putting this condition on the $g_a(x)$ and reducing the powers of x modulo n gives us a set of condition on the $c_i$.

† We take $g_a(x) = 0$ for $a = (0, 0, 0, \ldots, 0)$ and $g_a(x) = 1$ for $a = (1, 1, 1, \ldots 1)$.

$$\bullet \quad c_o^{\,2} = c_o \; ; \; c_{2i} = c_o^{\,2} \qquad i = 1, 2, \ldots, n-1.$$

Thus, to every vector $a \in V_n(F)$, there corresponds a polynomial $g_a(x)$ which can be written thusly;

$$g_a(x) = c_o + c_1 x + c_1^{\,2} x^2 + \ldots c_1^{\,2^{0(2)-1}} x^{2^{0(2)-1}}$$

$$c_{i_1} x^{i_1} + c_{i_1}^{\,2} x^{2i_1} + \ldots c_{i_1}^{\,2^{0(i_1)-1}} x^{2^{0(i_1)-1}}$$

$$\vdots$$

$$c_{i_{r-1}} x^{i_{r-1}} + \ldots c_{i_{r-1}}^{\,2^{0(i_{r-1})-1}} x^{2^{0(i_{r-1})-1}}$$

where

a) $i_1$ is the smallest integer $> 1$ such that $i_1 \neq 2^s$ modulo n for some integer s

b) $i_2$ is the next smallest integer $> i_1$ such that $i_2 \neq 2^s$, $i_2 \neq i_1 2^s$ mod. n

c) $i_3, i_4, \ldots, i_{r-1}$ are chosen in a similar fashion

d) $0(\ell)$ is the smallest positive integer such that $\ell\, 2^{0(\ell)} \equiv \ell \bmod n$. Note that for n a prime, $0(\ell) = 0(2)$ all $\ell$.

This leads to an immediate condition of the $c_i$

$$c_{i_j}^{\,2^{0(i_j)}} = c_{i_j} .$$

$c_{i_j}$ is contained in $GF(2^{0(i_j)}$ = the Galois field of $2^{0(i_j)}$ elements. Again for n a prime $c_{i_j} \in K$, the smallest field over F which contains the $n^{th}$ roots of unity. Thus $g_a(x)$ is parametrized by a set of independent

constants $c_o, c_1, c_{i_1}, \ldots, c_{i_{r-1}}$ where $c_o \in F$ and r is the number of irreducible factors of $x^n + 1/x + 1$ over F. The mapping of $a \rightarrow c_o, c_1, \ldots, c_{i_{r-1}}$ gives a linear map of $V_n(F)$ with a subgroup of the direct product of F with r copies of $K(FXK^r)$. For n a prime $V_n(F) \cong FXK^r$.

The coefficients $c_i$ may also be given via the Reed formula [1] [2]

$$c_k = \sum_{x^n=1} g_a(x)\, x^{-k}$$

$c_o = \Sigma a_i = 0$ indicates an even number of ones in a, $c_o = 1$ indicates an odd number of ones in a.

We shall henceforth write $g_a(x)$ as $g_a(c_o, c_1, \ldots, c_{i_{r-1}}; x)$ to illustrate the importance of the parameters $c_i$ in $g_a(x)$. Since we restrict x to the $n^{th}$ roots of unity, it is clear that the values of the $c_i$ tell the whole story†.

Let A be any (k, n) group code -- i.e., a k-dimensional subspace of $V_n(F)$. Then to A corresponds, via the isomorphism established, a subgroup G of the direct product $FXK^r$. If A is a cyclic code, then one or more of the $c_i$'s is identically zero -- and there are fewer parameters to range over [4]. Thus to any code, we may assign a set of $g_a$'s, functions of several variables restricted to subgroups of finite fields. The number of parameters or variables needed to describe the code will

† Note that this isomorphism depends on the particular choice of $\beta$.

determine for us the complexity of the code (for later purposes). The 4-7 cyclic code, for example, can be considered as corresponding to the values of $g_a(x)$ where

$$g_a(x) = c_o + cx + c^2x^2 + c^4x^4$$

where $c_o \in F$ and $c \in GF(2^3)$.

A fuller discussion of these facts and their applications is to be found in Solomon [3]. For purposes of the paper, we need only a representation of the general components of a vector a as the values of a polynomial $g_a(\beta^1)$, that is, the component $a_i$ is determined by the function of several variables $g(c_o, c_1, \ldots, c_{i_{r-1}}; \beta^i)$.

## 2. The General Formula

Let $a = (a_o, a_1, \ldots, a_{n-1}) \in V_n(F)$. The weight of a, $w(a)$, is defined to be the number of components in a equal to 1. For n odd, choose $\beta$ a fixed primitive generator of the $n^{th}$ roots of unity. We define the exponential weight of the vector a to be $\beta^{w(a)}$. This is a unique function of the weight except for the all-zero and all-one vector which yield the same value. We shall express $\beta^{w(a)}$ in terms of the parameters assigned to a in § 1. Since all group codes are expressible via these parameters, we shall have a general expression for the weights of code vectors.

$$f_i(a) = \beta^{a_i} = [1 + (1+\beta)\, a_i] \qquad i = 0, 1, 2, \ldots, n-1.$$

This is a function which is one for $a_i = 0$ and $\beta$ for $a_i = 1$. Let

$$f(a) = \prod_{i=0}^{n-1} f_i(a) = \prod_{i=0}^{n-1} (1 + (1+\beta)\, a_i).$$

The $f(a)$ is the desired exponential weight function.

Now to every $a \in V_n(F)$, there is assigned a polynomial $g_a(x)$ indexed by parameters $c_o, c_1, c_{i_1}, \ldots, c_{i_{r-1}}$. Therefore, we may write for $a_i$ in $f_i(a)$, $g_a(\beta^i)$. We thus have

$$f(a) = \prod_{i=0}^{n-1} (1 + (1+\beta)\, g_a(\beta^i)) .$$

Clearly

$$f(a) = \prod_{i=0}^{n-1} (1 + (1+\beta)) \, [c_o + c_1\beta^i + c_1{}^2\beta^{2i} + \ldots c_{i_1}\beta^{i_1} \ldots + c_{i_{r-1}}\beta^{i_{r-1}}].$$

The final formula is a function of the variables $(c_o, c_1, c_2, \ldots, c_{i_{r-1}})$.

To obtain the exact formula in simple terms, one must perform the prescribed multiplication (usually by machine). It is hoped that this formula will help shed some light on the distribution of weights in existing codes. Note that since any group code A is characterized by restricting the variables $c_i$ to lie in a certain subgroup of $FXK^r$, if the formula is worked out in detail, this may solve the problem.

## 3. Formula for Cyclic Codes

If A is a (k, n) cyclic code -- then, under the polynomial representation, at least one of the parameters vanish and the formula is reduced in complexity. If, in addition, the code A contains the all-one vector, we need only compute the even weight words -- i.e., setting $c_o = 0$ since all the odd weights are just (n-even) weights. Clearly, f(a), the exponential weight function, is a function of several variables and the computation thereof is usually non-trivial.

f(a) is a function of one variable in certain obvious cases -- i.e., if the cyclic code is generated by an irreducible polynomial (perhaps multiplied by $(1+x)$. For the special case where n is a prime p, $k = p + 1/2$, i.e., $r(p) = 2$[†] e.g., (4-7, 24-47, 9-17 ...), the formula has a very special form which we write below

$$f(c) = \prod_{i=0}^{p-1} \left(1 + (1+\beta)\left[\sum_{j=1}^{\frac{p-1}{2}} \beta^{i2^{j-1}} c^{2^{j-1}}\right]\right), \quad c \in GF(2^{p-1/2}).$$

This formula will yield the even-weight code words for the $(p+1/2, p)$ codes, a typical element of which is $a = (a_i)$ where $a_i = g_a(\beta^i)$ and $g_a(x) = \sum_{j=1}^{p-1/2} c^{2^{j-1}} x^{2^{j-1}}$. $\beta$ is a primitive $p^{th}$ root of unity.

Example 1: (4, 7)code

$$f(c) = \prod_{i=0}^{6} \left(1 + (1+\beta)\left(\sum_{r=1}^{3} \beta^{2^{j-1}} c^{2^{j-1}}\right)\right)$$

$$= 1 + (1+\beta)^4 c^7$$

$$= \begin{cases} \beta^4, & c^7 = 1 \\ 1, & c = 0 \end{cases}$$

† See [1], [3], and [4] for fuller discussion.

Analysis of f(a)

For these functions of one variable, one can easily see that $f(c) = f(c^2) = f(\beta^i c)$, $i = 1, 2, 3, \ldots, p-1$. Assuming $f(c) = \sum_{i=0}^{2^{p-1/2}-1} \gamma_i c^i$, one obtains condition on the $\gamma_i$,

$$\gamma_i = \gamma_{2i}, \quad \gamma_j = 0 \text{ for } j \not\equiv 0 \text{ modulo } p.$$

Thus there are at most $1 + \dfrac{\frac{2^{p-1/2}-1}{p} - 1}{p-1/2} = 1 + \dfrac{2^{p-1/2}-p-1}{p(p-1/2)}$ independent constants appearing in the final weight formula.

An algorithm for the actual computation of the formula for individual n is an interesting problem -- the solution of which may simplify the programming of the formula for functions of several variables. There is clearly much more work to be done in order to break down the formula into usable proportions.

GS:JEC

22 September 1961

## BIBLIOGRAPHY

[1] Mattson, H. F., and Solomon, G., "A new treatment of Bose-Chaudhuri codes," to appear in SIAM Journal, December 1961.

[2] Reed, I. S., and Solomon, G., "A decoding procedure for polynomial codes," Lincoln Laboratory Group Report 47-24.

[3] Solomon, G., "A new class of codes," Lincoln Laboratory Group Report 47G-0020.

[4] Solomon, G., "Quaternary cyclic codes," Lincoln Laboratory Group Report 47G-0022.